

CLAIMS

What is claimed is:

1. A data retrieval system for use with a data processing system, the system comprising:

a first memory;

a second memory accessible to said first memory;

a data file residing in said second memory, said data file containing stored data organized into nests;

a data structure residing in said first memory, said data structure designed to occupy a fixed amount of memory independent of content of said data file, said data structure organized according to hash values produced by a hash function for retrieving items in said data file, the hash values having associated offset values for accessing a nest of said data file; and

a data retrieval module in communication with said first memory, said data retrieval module operable to instantiate the hash function, to calculate a hash value based on input data, and to make an identification regarding a corresponding nest of the data file via said data structure, the identification based on the associated offset value of the hash value,

wherein the hash function is based at least in part on parameters selected according to characteristics of the data file, wherein the hash function is further designed to be optimized for content of said data file, and wherein the hash function is further designed to produce hash values based on the fixed amount of memory.

2. The system of claim 1, wherein said data retrieval module is further operable to load the corresponding nest from the second memory to the first memory, thereby resulting in a loaded corresponding nest residing within said first memory.

3. The system of claim 1, wherein said data retrieval module is further operable to search the corresponding nest of said data file for stored data matching the input data, and to retrieve the stored data.

4. The system of claim 2, wherein said data module is further operable to search the loaded corresponding nest for stored data matching the input data, and to retrieve the stored data.

5. The system of claim 4, wherein said first memory is a random access memory and said second memory is a disk memory.

6. The system of claim 1, wherein said stored data is compressed data, and wherein said data retrieval module is further operable to decompress the compressed data.

7. The system of claim 1, wherein said data structure has a data structure size based on a memory size of the first memory, and wherein said data file is organized into word nests of a number based on the data structure size.

8. The system of claim 1, wherein said data file has stored parameters, and wherein said data retrieval module calculates the hash value based on the stored parameters.

9. The system of claim 1, wherein said input data is a word of type string, and wherein said data retrieval module calculates the hash value based on at least one of characters parsed from the word and length of the word.

10. The system of claim 1, wherein the input data are further defined as a word of type string, and wherein the stored data are further defined as sound units for transcribing words of type string into audible speech, the sound units having associated words of type string.

11. The system of claim 10, wherein the sound units are further defined as phoneme combinations.

12. The system of claim 10, wherein the hash value is calculated based on character combinations parsed from the word of type string.

13. The system of claim 10, wherein the data file is encoded according to characters capable of being parsed from words of type string.

14. A method of constructing a data file for use with a data retrieval system of a data processing system, the data processing system having a first memory and a second memory, the method comprising:

choosing a data structure size for a data structure based on a memory size of the first memory;

organizing the data file into a number of nests based on the data structure size;

populating the data file with data based on a hash function and a plurality of parameters; and

storing said plurality of parameters within the data file.

15. The method of claim 14, the method further comprising:

repeatedly populating said data file with the data based on the hash function and the plurality of parameters;

varying the combination of parameters each time the data file is populated;

making an evaluation regarding a distribution of the data within the data file each time the data file is populated;

choosing a combination of parameters based on the evaluation; and

populating the data file with the data based on the combination of parameters,

wherein the plurality of parameters stored within the data file correspond to the combination of parameters.

16. The method of claim 14, wherein the data file is further defined as a lexicon database, wherein the data are further defined as sound units for transcribing words of type string into audible speech, the sound units having associated words of type string.

17. The method of claim 16, wherein the sound units are further defined as phoneme combinations.

18. The method of claim 16, wherein the hash function calculates a hash value based on character combinations parsed from the words of type string.

19. The method of claim 14, wherein the data file is encoded according to characters capable of being parsed from words of type string.

20. A data file manufactured according to the method of claim 14, the data file residing in memory operable with a data processing system.

21. A method of retrieving stored data based on input data for use with a data retrieval system of a data processing system, the method comprising:

receiving input data;

computing a hash value based on the input data;

determining an offset value based on the hash value, the offset value indicating a nest of a data file containing stored data, the data file organized into nests, the data file residing in a second memory accessible to said data processing system.

22. The method of claim 21, the method further comprising:

loading the nest from said second memory to a first memory accessible to said data processing system, resulting in a loaded nest within the first memory;

searching the loaded nest for matching stored data based on the input data; and

retrieving the matching stored data.

23. The method of claim 21, the method further comprising:

searching the nest for matching stored data based on the input data; and

retrieving the matching stored data.

24. The method of claim 21, wherein the first memory is a random

access memory, and wherein the second memory is a disk memory.

25. The method of claim 22, wherein the stored data is compressed,

the method further comprising:

decompressing the loaded nest within the first memory, resulting in a decompressed nest within the first memory, and

wherein said searching occurs within the decompressed nest.

26. The method of claim 21, wherein the input data are further defined as words of type string, and wherein the stored data are further defined as phoneme combinations for transcribing words of type string into audible speech, the phoneme combinations having associated words of type string.

27. The method of claim 21, wherein the hash value is calculated based on character combinations parsed from the word of type string.

28. The method of claim 21, wherein the data file is encoded according to characters capable of being parsed from words of type string.

29. The method of claim 21, wherein the data file has stored pluralities generated during construction of the data file, and wherein the hash value is calculated based on the stored parameters.

30. A transcription database system for use with a computerized transcription system implemented via a data processing system, the system comprising:

a random access memory accessible to said data processing system;

a disk memory accessible to said data processing system;

a lexicon file residing in said disk memory, said lexicon file containing compressed data corresponding to phoneme combinations for transcribing words of type string into audible speech, the phoneme combinations having associated words of type string, said lexicon file containing a stored combination of parameters generated during manufacture of said lexicon file;

a hash table residing in said random access memory, said hash table having a hash table size based on a memory size of said random access memory, said hash table organized according to hash values having associated offset values for accessing word nests of said lexicon file, said lexicon file organized into a number of word nests based on the hash table size; and

a data retrieval module in communication with said first memory, said data retrieval module operable to calculate a hash value for an input word of type string based on the stored combination of parameters, character combinations parsed from the input word, and a length of the input word, access a word nest of said lexicon file via said hash table, load the word nest into said random access memory, decompress the word nest, search the word nest for a word of type string matching the input word, and retrieve the phoneme combination associated with the word of type string.